DOCUMENT RESUME

ED 389 948 CE 070 518

AUTHOR Johnson, Scott D.; And Others
TITLE Understanding Troubleshooting Styles To Improve Training Methods.
PUB DATE 2 Dec 95
NOTE 17p.; Paper presented at the American Vocational Association Convention (Denver, CO, December 2, 1995).
PUB TYPE Speeches/Conference Papers (150) -- Reports - Research/Technical (143)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Equipment Maintenance; Identification; *Job Training; Logical Thinking; Operations Research; Postsecondary Education; *Problem Solving; Repair; Secondary Education; *Teaching Methods; *Troubleshooting; Vocational Education

ABSTRACT

An effective training program for preparing troubleshooters provides knowledge needed to understand the technology, teaches the process of troubleshooting, and provides the opportunity to practice using their knowledge and skill to diagnose faulty equipment. Recent research and common sense indicate many troubleshooting programs overemphasize theoretical knowledge and underemphasize knowledge and skills actually used while troubleshooting. Findings provide evidence that the use of functional flow diagrams during technical system instruction assists students in developing a more accurate conceptual understanding of the function of system components. The analysis of the verbal protocols reveals that most troubleshooters follow a similar process of troubleshooting: problem representation, fault isolation, and solution verification. The purpose of troubleshooting strategies is to help reduce the list of potential faults until the actual fault is identified. Strategies can serve two purposes: confirmation and exploration. Troubleshooters use various strategies to identify faults dependent upon factors such as level of expertise, type of system, and difficulty of the problem. Johnson (1991) has identified five common search strategies: trial and error, exhaustive, topographic, half/split, and functional. Although most troubleshooters rely on one or two favorite strategies, each strategy is useful under certain circumstances. (Contains 26 references.) (YLB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 389 948

Understanding Troubleshooting Styles to Improve Training Methods

Presented at the annual meeting of the American Vocational Association.
Denver, Colorado
December 2, 1995

Dr. Scott D. Johnson
Department of Vocational & Technical Education
University of Illinois at Urbana-Champaign

Dr. Jeffrey W. Flesher
Department of Workforce Education
Southern Illinois University

Mr. Shih-Ping Chung
Department of Vocational & Technical Education
University of Illinois at Urbana-Champaign

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☐ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
S. Johnson
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

Troubleshooting requires that technicians use their knowledge and skill to effectively interact with a complex technical system that is behaving in some unusual way. While some individuals seem to have a knack for developing troubleshooting skill, current research suggests that troubleshooting skill can be developed through properly designed instruction (Bedard & Frederiksen, 1992; Foshay, 1989). For example, studies have identified the critical knowledge base for successful troubleshooting (Keller, 1985; Kuipers & Kassirer, 1984; Morris & Rouse, 1986), troubleshooting strategies used by expert troubleshooters (Johnson, 1989, 1991; Rasmussen & Jensen, 1974), and recommended techniques for improving troubleshooting training (Foshay, 1989; Gott, 1988; Kieras & Bovair, 1984; Lesgold et al., 1988; Morris & Rouse, 1986; White & Frederiksen, 1987)

An effective training program for preparing troubleshooters is one that provides the knowledge needed to understand the technology, teaches the process of troubleshooting, and provides students with the opportunity to practice using their knowledge and skill to diagnose faulty equipment. Instructors have difficulty in each of these three critical areas of training. Developing understanding of the technology is becoming more difficult each day as technology becomes more complex. Teaching the process of troubleshooting is difficult because that process is extensively cognitive in nature and therefore is hidden from view in the mind of the troubleshooter. Providing opportunities to practice troubleshooting in a controlled instructional environment is difficult because of the lack of equipment for training purposes, limited availability of tooling, problems with wear and tear inherent in the process of disassembly and assembly, and the increased time necessary to physically do all of the activities essential for practical training.

Over the past eight years we have conducted in-depth studies of the cognitive and behavioral processes used by technical troubleshooters. Our subjects have ranged from post-secondary students who are studying to become technicians to expert troubleshooters in industrial settings. Some of our studies have involved the administration of technical tests to identify knowledge difference, others have examined the influence of different technical illustrations, and others have involved providing our subjects with faulty equipment and asking them to "think aloud" as they try to locate the fault. Using these "verbal protocols," we are able to learn more about the troubleshooter's initial problem space representation, the strategies used to locate faults, and the cognitive process used to troubleshoot.

Through this in-depth study of over fifty troubleshooters in a variety of technical areas, we have begun to develop a better understanding of these issues. The purpose of this paper is to share some of our insights about troubleshooting expertise and to suggest how troubleshooter training can be

CE 070 518

BEST COPY AVAILABLE

modified to address the difficulties that are inherent in such training. Our discussion will be organized around the issues of course content and troubleshooting strategies.

What Should be Taught in a Troubleshooting Course?

Domain Content or Theoretical Knowledge

Recent research and common sense tell us that many troubleshooting programs overemphasize theoretical knowledge and place too little emphasis on the knowledge and skills that are actually used while troubleshooting. Trainers emphasize "nice to know" information such as equation solving and theory learning rather than focusing on the "must know" information needed by technicians. For example, students of electronics often study atomic theory, spend considerable time learning equations to calculate circuit conditions (e.g., Ohm's law, Kirchhoff's laws), and apply those equations to solve circuit design problems. Do we really mean to teach system functions from a design perspective? The emphasis placed on design in many troubleshooting courses is not surprising when you consider that we "teach as we were taught" and many of the technical instructors have either been trained as engineers or were taught by engineers, a field where design is a major focus of instruction. A focus on design is appropriate if we are preparing developing engineers, but not if we are preparing technicians who will troubleshoot systems, not design them.

Flesher (1993) conducted an in-depth protocol analysis of electronic troubleshooters from three different contexts: design, production, and repair. Flesher provided technicians from each of these contextual settings with a faulty electrical system and analyzed their troubleshooting performance from a cognitive perspective. His results showed that context influenced the troubleshooters' initial frame of reference, which impacted their ability to locate faults. Similar evidence of the influence of context on performance has been noticed in Johnson's study of generator troubleshooters (1987) and Martin and Beach's study of CNC machining (1992). Martin and Beach, for example, noticed differences in the thinking patterns of technical personnel as a result of prior experience and the type of training they received. They also noticed that when they were confronted with a technical problem, engineers thought about economic concerns while machinists thought about contingencies and setup people thought about practical matters.

Although most technical instructors emphasize basic electronic and mechanical theory in their courses, our studies suggest that possessing theoretical understanding is not a significant predictor of troubleshooting ability. One recent study provides an example (Johnson, Flesher, Jehng & Ferej, 1993). The subjects in this study were university students who have completed an advanced avionics electrical systems course. Performance on a set of practical troubleshooting exercises was used to sort the subjects into two groups (i.e., high and low performers). The mean scores on a electrical system post-test examination was 19.82 for the high performing troubleshooter group (SD = 1.94) and 19.88 for the low performing group (SD = 2.13) out of a total score of 24. This slight difference in mean scores was not statistically significant, $t(32) = .073$, $p > .05$. These findings suggest that theoretical knowledge of systems may not be as important for troubleshooting as is commonly thought.

Developing System Understanding

Trainers should also be sensitive to the experience and capabilities of their students when selecting content for troubleshooting courses. Trainers typically use complex schematic diagrams to teach about the structure, function, and operation of technical systems. This approach is fine for students who have considerable experience with the abstract diagrams, but if the students lack the prerequisite knowledge need to comprehend the diagrams, the training will be confusing and frustrating for the students. This problem is especially evident when students with extensive mechanical experience return for training in electronics. When students do not have the prerequisite

background knowledge needed to comprehend the abstract diagrams, they are unable to develop a conceptual understanding of the system. A simpler, more conceptual drawing of the technical system needs to be provided as an introduction to the system. By using simpler diagrams, students will be able to see the subsystems within the total system and have a better understanding of the relationships between the various components. Once the students have learned about the basic layout, function, and operation of the system, it is then possible to use the more abstract and complex schematics for instruction.

Research has shown that conceptual understanding of systems is important (Chi, Feltovich & Glaser, 1981; Larkin, McDermott, Simon & Simon, 1980). In the case of electronics instruction, White and Frederiksen (1987) suggest that instructors help students acquire a *circuit schema* before introducing theories and mathematical calculations. Brown and Burton (1987) also advocate teaching novice technicians conceptual knowledge that is based on qualitative, causal models of how systems function.

Technical instruction typically relies on schematic diagrams to illustrate system structure and operation. Even though schematic diagrams are used widely as a visual aid for technical instruction, it has been suggested that they are often too complex and abstract to be effective for initial system instruction (Johnson, 1991). A recent study tested the effectiveness of conceptual diagrams for teaching initial system understanding (Johnson & Satchwell, 1993). A commercial individualized training manual was used by one group university aviation students to learn about small aircraft electrical systems. This manual uses schematic diagrams to teach about electrical systems and subsystems and introduces the major electrical concepts of DC generation, dual-bus DC distribution, multiple-bus DC distribution, and AC generation/distribution. An example of a schematic from the manual is shown in Figure 1.

The second group was given a modified training manual in which each schematic diagram in the manual was preceded with a functional flow diagram showing the system at a conceptual level and indicating the causal nature of the system. Figure 2 illustrates the functional flow diagram that corresponds to the schematic shown in Figure 1 and is typical of those used throughout the "conceptual" version of the manual.

One unit of the training manual was assigned each week over a four week period. The subjects were asked to view a videotape that accompanied the manual, read each unit of instruction, and answer the questions at the end of each unit. Following each unit of instruction, the subjects were given unit tests that were designed around the structure, function, and behavior of the system. To provide additional depth of analysis, four subjects from each group were randomly selected to perform several card sorting tasks after completing each unit test.

The results showed that the use of concept maps had little effect on the subjects' understanding of the system structure and component function, but had a significant effect on their level of behavioral understanding (Mann-Whitney U test statistic = 2.50, $p < .01$). This finding suggests that the students who learned from the training manual containing the functional flow diagrams achieved greater understanding of the behavior of the systems than the group that learned from the manual containing only schematic diagrams.

The card sorting tasks identified a significant difference between the two groups in their ability to develop accurate knowledge structure maps ($t_{(82)} = -4.705$, $p < .01$). This finding shows that students who used functional flow diagrams created more expert-like mental models of technical systems than did students who used only schematic diagrams. There was also a clear difference between the two groups in their ability to place the concept cards in the correct location on the knowledge structure maps. The group that learned from the functional flow diagrams was able to accurately link 81% of the concepts while the group that learned from the schematic diagrams could only link 52% of the concepts correctly. This finding suggests that using functional flow diagrams

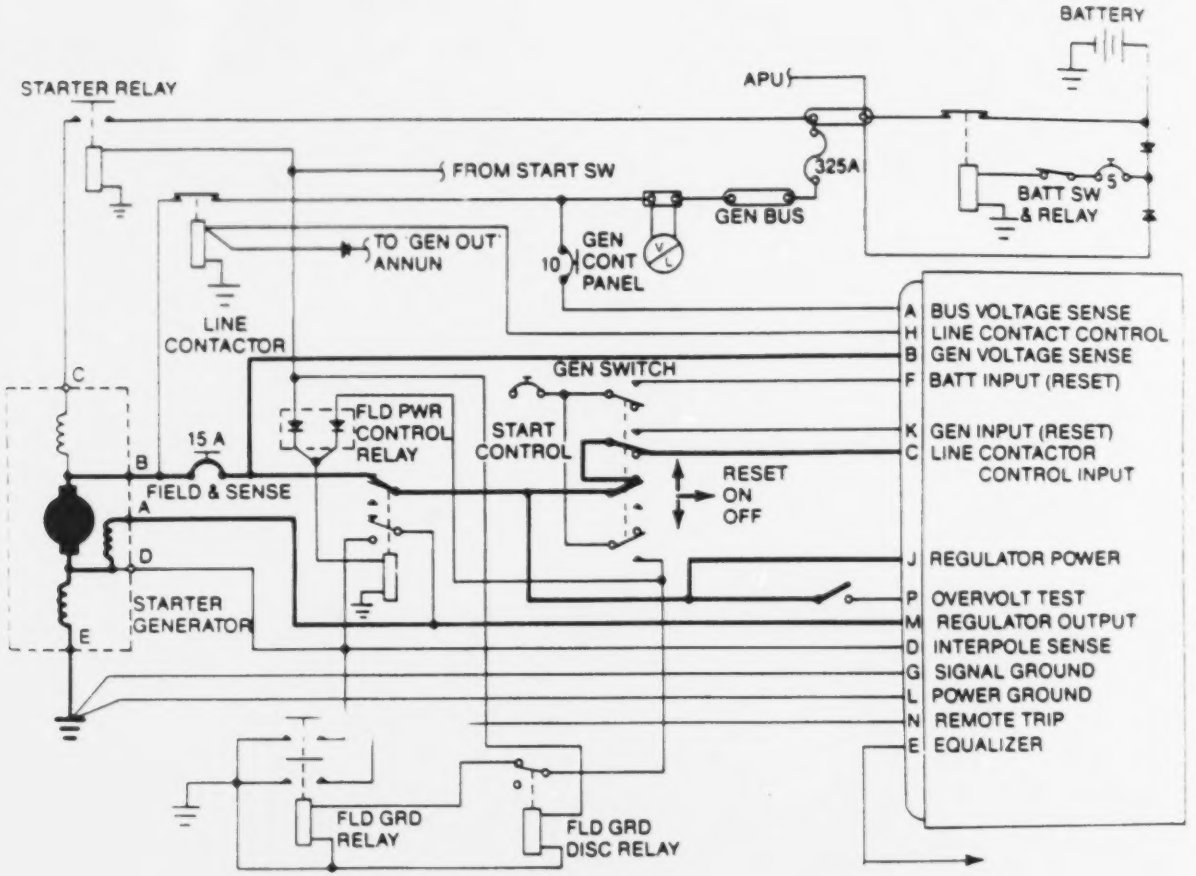

Figure 1. An example of a type of schematic diagram.

Note: Taken from Introduction to King Air Electrical Systems, Field Training Package. (pp. 3-4) by Beech Aircraft Corporation, 1986, Wichita, KS. Reprinted with permission.

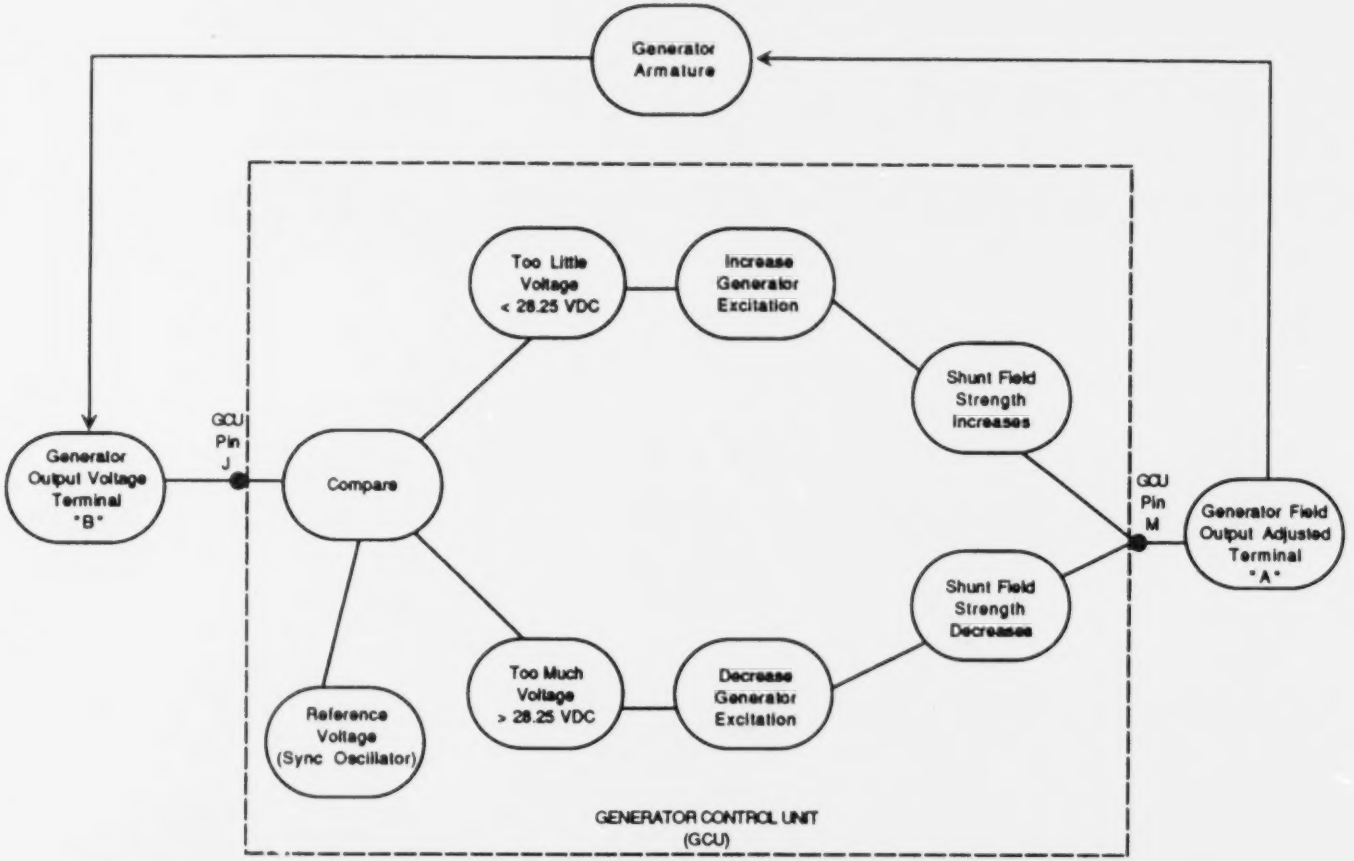

Figure 2. An example of a functional flow diagram (Johnson, & Satchwell, 1993).

BEST COPY AVAILABLE

during technical instruction enhances the student's ability to develop more accurate knowledge structures. Overall, these findings provide evidence that the use of functional flow diagrams during technical system instruction assists students in developing a more accurate conceptual understanding of the function of system components.

What Processes and Strategies are used by Troubleshooters?

The analysis of the processes and strategies used by troubleshooters was based on the verbal protocols collected from troubleshooters as they attempted to locate faults in technical systems. See Flesher (1993), and Johnson (1987, 1988, 1989) for specific descriptions of the data collection and analysis methods that were used. The analysis of the verbal protocols reveals that most troubleshooters follow a similar process of troubleshooting. This process generally involves three phases that constitute effective troubleshooting: (a) problem representation, (b) fault isolation, and (3) solution verification. The following discussion addresses each of these troubleshooting phases and concludes with a more detailed discussion of the troubleshooting strategies used and the types of information accessed within the various phases.

Troubleshooting begins in response to a symptomatic condition. That condition can be as broad as simply recognizing that an electronic system is not working, or as specific as noticing an obvious or repetitive fault. The troubleshooter then gathers information to support the development of an initial hypothesis. Once enough information has been gathered to support an initial hypothesis, the troubleshooter enters phase two, fault isolation. Through the process of hypothesis generation and evaluation the problem solver reduces the size of the problem space. This is a cyclic process with each pass through the model generating a closer approximation of the fault. Troubleshooting strategies play a key role in this process, allowing the troubleshooter to reduce multiple hypotheses to identify the actual fault. It is important to note that when the troubleshooter determines that the current hypothesis is not the fault, the troubleshooter returns to the problem representation phase to either "reshape" the problem space by acquiring new information, or takes a broader view of the problem. When the problem space has been reduced to a likely fault, the third phase of the model, solution verification, begins. Verification can be accomplished through additional tests or component replacement. Troubleshooting does not end until the fault can be verified.

Problem Representation

The first phase of the troubleshooting process, problem representation, includes the initial evaluation of a system and any assumptions brought to the problem solving situation. These early efforts establish a broad definition of the task environment. Based on this general search, most troubleshooters have little problem verifying that a fault does exist in a system. It appears that troubleshooters often bring baseline assumptions based on past experience into new situations which help to frame the novel activity. These assumptions generally promote efficiency but may also generate confusion in poorly matched tasks. Maintenance technicians may not search for design flaws, particularly in systems which they know have previously been fully functional. The assumption of design integrity that this knowledge enables serves to restrict the likely problem space to component failures to age, wear, or environmental influences (Flesher, 1993).

The initial mental model, or problem space representation is very important. Bartlett (1958) observed that problem solvers often established a reference that they seldom question. In a study of students' aircraft electrical system troubleshooting performance, both skilled and less-skilled subjects conducted an initial evaluation that identified a limited number of subsystem faults. The subjects rarely considered faults in other subsystems not identified in the initial search. Even with cues, such as "is that all of the problems in the board", subjects rarely questioned the initial reference frame they had established (Johnson et al., 1993). The quality of initial models influences the subsequent information acquisition efforts. Schlager, Means, and Roth (Schlager, Means & Roth, 1988) observed expert and novice troubleshooting in a complex system. The expert was able to focus on a small

section of the overall system, while the novice, given the same symptoms, became enmeshed in an overwhelming amount of detail resulting in an inability to identify the correct problem space.

The quality of the initial frame of reference also reflects the troubleshooters specific past experiences with a particular equipment or system. The process of troubleshooting familiar equipment can be very different from the process used on unfamiliar systems. On systems that are often encountered, the process may be one of simply symptom/fault recognition rather than a fully developed troubleshooting activity. In the study conducted by Johnson et al. (1993) aircraft maintenance technicians were asked to validate a problem set established for student subjects. One maintenance technician with more than thirty years experience was touted by the group as the best troubleshooter due to his ability to generate immediate solutions to problems encountered on a daily basis. During his efforts to troubleshoot an unfamiliar system in the research environment, he was unable to demonstrate his expertise, even though the system contained familiar components. In fact, his troubleshooting process appeared very weak. It is likely that over the many years of activity in a stable environment this technician had memorized the potential problems and no longer had to troubleshoot to identify faults. Flesher (Flesher, 1993) noticed that maintenance and production technicians appeared to ignore design flaw possibilities, relying on a common frame of reference established through experience with equipment that had valid designs.

The net result of past experiences, frames of reference, or any initial information that has been provided, is a basic problem representation or initial problem space. This can easily range from a very limited set of components or a single component based on the recognition of symptoms, or it can be as broad as simply establishing that the system is not operating correctly. Although Rasmussen and Jensen (1974) describe the process of troubleshooting as a search through a hierarchy of units ranging from subsystems, stages, and finally components, the use of strategies appears to be limited during the initial phase to functional evaluations that are based on previous system knowledge, sensory checks, and system operation. As a result, the initial frame of reference is based primarily on a symptomatic conditions.

Fault Isolation

Once the troubleshooter has established that a problem exists the process of fault isolation begins. This activity is accomplished within the framework of the hypothesis generation and testing cycle. Our subjects' protocols demonstrated that the "generate and test" method of problem solving is the dominate procedure. When we compared the number of "generate and test" cycles (we called these cycles *episodes*) needed by high and low performers to solve a problem, there appeared to be no difference. Virtually all of the troubleshooters relied on this process. When we examined the ability of the troubleshooters to correctly decide if the potential fault they were considering was the actual fault, the subjects appeared to rely on various types of information to make their decision. This evaluation process is used to reduce the size of the problem space as the troubleshooter eliminates potential faults from consideration. While there was no difference in the average number of episodes for high and low performers, there were significant differences in their ability to correctly interpret the meaning of the information they collected to evaluate the proposed faults. Higher level performers had fewer misinterpretation errors and were more likely to "recover" from their misinterpretations.

The studies referenced earlier describe four discrete levels of system related problem space definition and hypotheses; (a) system, (b) subsystem, (c) device, and (d) component. A system level hypothesis is one that does not reduce the problem space beyond the entire equipment or system under investigation. An example of this type of hypothesis would be the general statement of, "I think something is wrong", or for example within an automotive system, "The car doesn't start." A troubleshooter may understand that before the engine will start there must be fuel available, spark, and engagement of the starter motor. Looking at the gas gauge provides an indication of the fuel system and listening for the starter motor provides a clue to the operation of the electrical system. A

subsystem level hypothesis reduces the likely problem space to a discrete subsystem within the larger system. An example of this type for our automobile scenario would be, "maybe something is wrong with the electrical system." A device level hypothesis further deduces the problem space to a limited number of components within a subsystem. Using the same analogy an example could be, "it's probably either the battery, starter, or solenoid." A component level hypothesis is the most specific level and results in a single component being identified as the potential cause of failure. The statement, "I think the battery is dead" is an example of this level.

Solution Verification

Although, we know it is important, based on our collected verbal protocols, not every troubleshooter verifies their final troubleshooting decision. Except for those troubleshooters who failed to identify the faults, most troubleshooters stop troubleshooting soon after they think they found the fault. The subjects who verified their previous decision were more likely to be those who troubleshoot successfully. We are not suggesting that the three phases go linearly each time. Actually, whether successful or not, the troubleshooters verified their results quite often. Even the low performers tried out different ways (as best they could) to verify their unexpected findings throughout the process of troubleshooting a faulty system. Being able to verify a previous decision indicates that the problem solver knows more than one way to complete the task. Without a certain amount of domain knowledge and skill, the troubleshooters cannot perform the solution verification phase efficiently. That was the case in the low performer's verbal protocols. Quite a lot of irrelevant and incorrect testing was done to the validate collected information.

Repetitive testing is another form of solution verification, although, it indicates the non-readiness of the performer. Either the troubleshooter was not ready for the previous troubleshooting activity, or is unfamiliar with the system function that makes them repeat the exact testing again. Usually, verifying a verification test helps the troubleshooter refine or narrow the problem space and is a strong indicator that the troubleshooter knows if they are on the right track.

Troubleshooting Strategies

Strategies play the key role in managing this phase of the problem solving process. The purpose of troubleshooting strategies is to help reduce the list of potential faults until the actual fault is identified, in other words, the size of the initial problem space is reduced. Strategies can be serve two purposes; confirmation and exploration. Strategies may be used to confirm a hypothesis that presupposes a condition. The statement of "I think there is an open line in this subsystem" presupposes that the open circuit condition exists within the system under investigation. A strategy can be used in this situation to reduce the area in which the open is believed to be located as well as confirm its existence. Strategies may also be exploratory providing new information to the troubleshooter that is not based on a presupposition of specific circuit condition. Probably the best known method of exploratory strategy use is simple trial and error. The troubleshooter applies information gathering techniques to a limited component set in an effort to provide some addition information, if not the solution itself.

A strategy is the method of managing the acquisition of information in a systematic manner in order to identify and test potential faults. The measurement of a voltage at a certain point in a circuit that has been identified as the output of a non-functioning unit is not an example of a strategy, and in fact strategies are independent of specific information gathering techniques. A half-split strategy can be accomplished through voltage or continuity checks with a multimeter, by signal tracing with an oscilloscope, or from multiple occurrences of tactile inspection. Trial and error is still a systematic process when directed to a level of specificity or limited component set.

Troubleshooters use various strategies to identify faults dependent upon factors such as; the troubleshooter's level of expertise, the type of system, and the difficulty of the problem. Johnson (1991) identified five common search strategies (a) trial and error, (b) exhaustive, (c) topographic, (d) half/split, and (e) functional. Trial and error is commonly used by novices. It is a random search rarely used by experts. Lesgold et al. (1986) provided another version of this strategy called "swaptronics" in which parts or subsystems known to be good, are inserted until one identifies the correct area or component. Exhaustive searches require little expertise, but are efficient only in situations where the component set is small. Topographic searches are directed by tracing a schematic through the system identifying a series of good/bad checks (Rasmussen & Jensen, 1974). Two types of topographic search strategies have been observed in troubleshooting activity based on the starting point of the search. The first is starting at a known good point and working toward the fault (forward topographic), and the second is starting from the faulty condition (i.e., non-working lamp), and working toward the input (reverse topographic) (Johnson et al., 1992). The half/split, or binary search method is an efficient means of reducing a problem space by checking for a proper condition at the middle of the identified problem space. If a proper condition exists a new reference point is selected in the middle of the remaining problem space. The process is repeated until a single component remains. The functional method is the most difficult. Information about the system and its components forms the basis of hypothesis selection (Johnson, 1991). Various types of functional search exist as well. Basing a hypothesis on known system behavior and expected outcomes is the basic type of functional search. Problem solvers can also construct a "runnable" mental model of the system in which the fault conditions are simulated (deKleer & Brown, 1981) although Norman (1983) observes that generally the ability to run mental models is very limited.

While most troubleshooters rely on one or two favorite strategies, each strategy is useful under certain circumstances. Topographic, exhaustive, and trial and error searches are selected because little cognitive effort is needed. Some methods, such as the half/split, are selected because they are efficient at eliminating a large number of possibilities. Other reasons for selecting methods include their ease of use, their low cost in terms of time and materials, and their reliance on the availability of spare parts and other resources.

The collected protocols from the two studies reveal a primary attribute of skilled performance and strategy use. There was no apparent difference in the types of strategies used by high and low level performers . In most cases both groups of troubleshooters have learned to use various strategies through formal instruction. Although these was no significant difference in the types of strategies used, skilled troubleshooters were able to use multiple strategies and change strategies when necessary. Less skilled subjects in the study conducted by Johnson et al. (1993) relied on a very limited strategic approach. These subjects were effectively *strategy bound*, unable to successfully troubleshoot due to their limited use of multiple strategies. Conversely, skilled performers were able to apply multiple approaches, even reverting to a mature form of trial, and error when unable to gain additional information. The highest performing troubleshooters are masters of many different troubleshooting strategies. By selecting a very limited component set for experimentation (manipulation of wires, replacement of components) the troubleshooters were able to successfully complete the fault isolation process.

The process of troubleshooting requires an integrated ability to collect, process, and evaluate external and internal information. A correct fault solution may be obtained after many instances of incorrect hypothesis evaluation, or information interpretation. Conversely, an incorrect judgment may result from a single failure to correctly interpret acquired information or evaluate generated hypothesis. This implies that an important element in the process is the ability to continue cycling through the hypothesis generation and evaluation process and to verify the potential solutions (Johnson et al., 1993).

Types of Information Accessed

Troubleshooters rely on information to learn about the systems and its faults. This information is acquired from either internal or external sources (see Figure 3). As an internal source, the individual's long term memory contains both declarative knowledge and procedural knowledge. The external sources of information could include job aids, technical support, technical evaluations, and sensory evaluations. After acquiring information, the troubleshooter interprets the accuracy and relevance of the information and processes it by relating the information to the current situation and to their past experience.

There appears to be a few interesting difference between expert and novice troubleshooters. Experts tend to seek information through technical means more than novices while novices seek information through sensory means more than experts. In addition, there appears to be little difference between experts and novices in their use of job aids and other technical supports.

Following its acquisition, the information must be interpreted by the troubleshooter. There appears to be little difference in the level of information interpretation between expert and novice troubleshooters. There is also little difference in the ability of the troubleshooters to interpret information. This lack of difference to accurately interpret information can be explained by the fact that people typically do not seek information that they are unable to understand. While expert and novice troubleshooters certainly gather different types of information, each group is equally able to interpret what they collected. However, if the novices were given the information that was gathered by the experts, it is doubtful that the novices could have accurately interpreted that information.

Implications for Training Troubleshooters

Given the results of our research, what can we recommend about teaching troubleshooting in technical training programs? While additional research is needed to validate our initial results, we are comfortable making the following recommendations.

1. Ensure that students acquire a clear conceptual understanding of system structure, function, and operation of the systems they will be working on. Conceptual illustration such as functional flow diagrams seem to be useful for developing conceptual system understanding.

2. Help students develop competence in a wide variety of troubleshooting strategies.

3. Explicitly teach students to follow the "generate and test" process of troubleshooting. Besides knowing how to follow the troubleshooting process, they should be aware of what they are doing and why it is important.

4. Provide extensive shop, laboratory, and field experiences in order to develop the consistent patterns of behavior that are associated with expert troubleshooters.

5. Emphasize the importance of using the senses to obtain initial problem information and have enough practice to become proficient at acquiring information in that manner.

6. Provide extensive practice to develop good system and circuit tracing skills. Tracing strategies such as the half/split method of system analysis must be explicitly taught. While this is a well recognized strategy of troubleshooting among experts, we have been surprised by the number of troubleshooters who do not use it and disappointed by the number of trainers who do not directly teach this strategy.

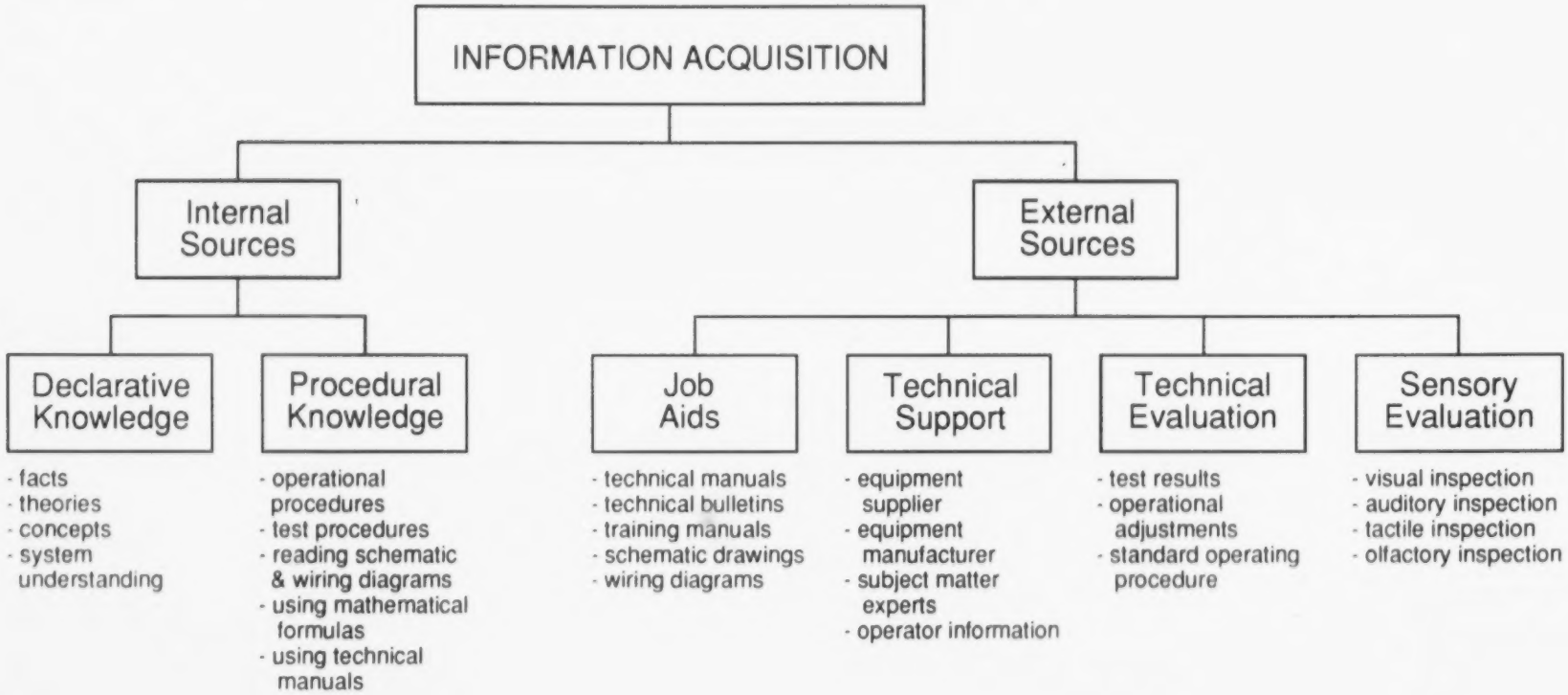

Figure 3. Sources of Information for Troubleshooters.

7. Explicit instruction in the use of diagnostic heuristics or algorithms is necessary to develop good troubleshooters (Morris & Rouse, 1986). Future technicians should be instructed in the use of multiple strategies, encouraged to explore alternative methods, and receive explicit instruction in four key areas: (a) problem space management, (b) the hypothesis generation/evaluation cycle, (c) the application of strategies, and (d) information acquisition methods (test equipment and sensory based inspections).

References

Bartlett, F. (1958). *Thinking: An experimental study*. New York: Basic Books.

Bedard, D., & Frederiksen, C. H. (1992, April). *Comprehension and use of knowledge by students in learning complex technical skills*. Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

Brown, J. S., & Burton, R. R. (1987). Reactive learning environments for teaching electronic troubleshooting. In W. B. Rouse (Ed.), *Advances in man-machine systems research*, (pp. 65-98). Greenwich, CN: JAI Press.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5*, 121-152.

deKleer, J., & Brown, J. S. (1981). Mental models of physical mechanisms and their acquisition. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*, (pp. 285-309). Hillsdale, NJ: Erlbaum.

Flesher, J. W. (1993). An exploration of technical troubleshooting expertise in design, manufacturing, and repair contexts. *Journal of Industrial Teacher Education, 31*(1), 34-56.

Foshay, W. R. (1989). What we know (and what we don't know) about training of cognitive strategies for technical problem-solving. *Journal of Structural Learning, 10*, 111-125.

Gott, S. P. (1988). Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies. In E. Z. Rothkopf (Ed.), *Review of research in education*, (Vol. 15, pp. 97-109). Washington DC: American Educational Research Association.

Johnson, S. D. (1987). *Knowledge and skill differences between expert and novice service technicians on technical troubleshooting tasks* (Technical Report #20): St. Paul: University of Minnesota, Training and Development Research Center. (ERIC Document Reproduction Service No. ED 284 054).

Johnson, S. D. (1988). A cognitive analysis of expert and novice troubleshooting performance. *Performance Improvement Quarterly, 1*(3), 38-54.

Johnson, S. D. (1989). A description of expert and novice performance differences on technical troubleshooting tasks. *Journal of Industrial Teacher Education, 26*(3), 19-37.

Johnson, S. D. (1991). Training technical troubleshooters. *Technical and Skills Training, 27*(7), 9-16.

Johnson, S. D., Flesher, J. W., Jehng, J.-C., & Ferej, A. (1993). Enhancing electrical troubleshooting skills in a computer-coached practice environment. *Interactive Learning Environments, 3*(3), 199-214.

Johnson, S. D., & Satchwell, R. E. (1993). The effect of functional flow diagrams on apprentice aircraft mechanics' technical system understanding. *Performance Improvement Quarterly, 6*(4), 73-91.

Keller, R. (1985). *Human troubleshooting in electronics: Implications for intelligent maintenance aids* (AFHRL Tech. Paper 85-34): Denver, CO: University of Denver, Denver Research Institute, Social Systems Research and Evaluation Division.

Kieras, D. E., & Bovair, S. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8*(3), 255-273.

Kuipers, B., & Kassirer, J. P. (1984). Causal reasoning in medicine: Analysis of a protocol. *Cognitive Science, 8*(4), 363-385.

Larkin, J., McDermott, J., Simon, D. P., & Simon, H. A. (1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335-1342.

Lesgold, A., Lajoie, S., Bajpayee, J., Bunzo, M., Eastman, R., Eggan, G., Greenberg, L., Logan, D., McGinnis, T., Weiner, A., & Wolf, R. (1988). *A computer coached practice environment for the*

manual test station shop: Sherlock's influence on trainee job performance. Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.

Lesgold, A., Lajoie, S., Eastman, R., Eggan, G., Gitomer, D., Glaser, R., Greenberg, L., Logan, D., Magone, M., Weiner, A., Wolf, R., & Yengo, L. (1986). *Cognitive task analysis to enhance technical skills training and assessment* : Pittsburgh, PA: University of Pittsburgh, Learning Research and Development Center.

Martin, L. M., & Beach, K. (1992). *Technical and symbolic knowledge in CNC machining: A study of technical workers of different backgrounds* (MDS-146). Berkeley, CA: University of California at Berkeley: National Center for Research in Vocational Education.

Morris, N. M., & Rouse, W. B. (1986). *Review and evaluation of empirical research in troubleshooting* (Interim Report NPRDC TR 86-20): San Diego, CA: Navy Personnel Research and Development Center.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. Stevens (Eds.), *Mental models*, (pp. 7-14). Hillsdale, NJ: Erlbaum.

Rasmussen, J., & Jensen, A. (1974). Mental procedures in real-life tasks: A case study of electronic trouble shooting. *Ergonomics, 17*(3), 293-307.

Schlager, M., Means, B., & Roth, C. (1988, April). *Cognitive analysis of expert knowledge: Input into design of training*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

White, B. Y., & Frederiksen, J. R. (1987). *Causal model progressions as a foundation for intelligent learning environments* (Technical Report #6686): BBN Laboratories, Cambridge, MA.